## REMARKS

In the present Office Action claims 1-5 were examined. Claims 1-5 were rejected and objected to, and no claims were allowed. By this amendment, claims 1-5 have been amended, no claims were cancelled and claims 6-16 were added. Accordingly, claims 1-16 are presented for further examination.

### Objection and Rejection under 35 U.S.C. §112, Second Paragraph:

In Sections 1-3 of the Office Action the Examiner objects to and rejects, under 35 U.S.C. §112, second paragraph, claims 1-5. In particular, the Examiner objects to the recitation of the term "Specifying" in claim 1 and notes insufficient antecedent basis for a limitation "said identified data elements" in claims 1-5.

Claims 1-5 have been amended with the Examiner's comments in mind. In view of the amendments, it is respectfully requested that the Examiner reconsider and withdraw the objection and rejection to claims 1-5 under 35 U.S.C. §112, second paragraph.

### Rejections under 35 U.S.C. §103:

In Section 5 of the Office Action the Examiner rejects claim 1 under 35 U.S.C. §103(a) as being unpatentable over Sheffield (U.S. Patent No. 5,832,431) in view of Goldberg (U.S. Patent No. 5,907,847). This rejection is respectfully disagreed with, and is traversed below.

Sheffield describes a database interface and interface customization tool implemented as a graphical user interface (GUI). The GUI may be manipulated by an applications programmer (e.g., formatting SQL statements) to select and retrieve desired data from tables in an application database (see Abstract, and Col. 3, lines 8-23). In summary, Sheffield appears to describe a conventional database management interface.

As noted by the Examiner, Sheffield does not describe or suggest "generating executable code from said specified data elements for extracting said specified data elements from said operational database" as recited in claim 1, as originally file and as now written.

The Examiner attempts to cure this deficiency by a proposed combination of Sheffield and Goldberg.

Goldberg describes a database management system (DBMS) that permits coupling of an object's state and behavior information to the object definition stored within the DBMS. After quoting relevant portions of Goldberg defining an object's state and behavior the Examiner concludes that it would be obvious to modify the data extraction method of Sheffield to include the generation of executable code for extracting identified data elements from the database. This conclusion is respectfully disagreed with, and traversed below.

Goldberg is not seen to describe or suggest that an object's behavior (defined as operations performed with the object) includes operations for retrieving the object from the DBMS storing the object. On the contrary, Goldberg describes the need for a mechanism to invoke or initiate behavior (see Goldberg at Col. 7, line 64 to Col. 8, line 7). Additionally, at Col. 4,lines 44-50, Goldberg recites:

> "A client process having a mechanism for initiating behavior (i.e., a behavior initiator) accesses the DBMS server to obtain an object having both state and behavior. Once retrieved, the mechanism is used to invoke the object's behavior. The mechanism is a language interpreter or a just-in-time (IT) complier, for example."

Accordingly, Goldberg does not describe or suggest that an object's own behavior includes operations for "generating executable code from said specified data elements for extracting said specified data elements from said operational database" as recited in claim 1, as originally file and as now written. As noted in the above portion of Goldberg, the document describes use of a third party utility to invoke an object's behavior.

It is respectfully submitted that there is no motivation, suggestion or incentive for one skilled in the art to combine the cited documents as is suggested by the Examiner. However, even if the proposed combination of Sheffield and Goldberg were made, the combination still does not describe or suggest all of the limitations of claim 1 as filed and as now written. For example, the proposed combination would describe a database interface and interface customization tool implemented as a GUI, the GUI permitting an applications programmer (e.g., formatting SQL statements) to select and retrieve desired data from tables in an application database (Sheffield), the desired data including state and behavior information that can be invoked once retrieved from the data store (Goldberg).

In view of the foregoing, it is respectfully submitted that independent claim 1 is clearly patentable over the Examiner's proposed combination of Sheffield and Goldberg. Accordingly, the Examiner is respectfully requested to reconsider and remove the rejection of claim 1 under 35 U.S.C.§103(a).

In Section 6 of the Office Action the Examiner rejects dependent claims 2-5 under 35 U.S.C. §103(a) as being unpatentable of Sheffield in view of Goldberg as applied to claim 1, and further in view of Medl (U.S. Patent No. 6,108,004). This rejection is respectfully disagreed with, and is traversed below.

The Examiner states that Sheffield as modified (e.g., the proposed combination with Goldberg) describes that data may be extracted and stored in another location and uses Medl to describe that the other location may be a staging database. The Examiner further notes that Sheffield as now modified describes all of the limitations (e.g. relational databases and use of a GUI) of the dependent claims 1-5.

It is respectfully submitted that Medl describes an interface including a sequence of GUI templates arranged for assisting a user develop data mining objects. Accordingly, Medl is not seen to cure the above noted deficiency in the proposed combination of Sheffield and Goldberg. For example, Medl is not seen to expressly or implicitly describe or suggest "generating executable code from said specified data elements for extracting said specified data elements from said operational database" as recited in claim 1, as originally file and as now written.

Since independent claim 1 is clearly patentable over the proposed combination of Sheffield, Goldberg and Medl, then claims 2-5 that depend from and further limit independent claim 1 are also patentable over the proposed combination. Therefore, the Examiner is respectfully request to reconsider and remove the rejection of claims 2-5 under 35 U.S.C. §103(a).

As a part of this response claims 6-16 have been newly added. It is asserted that these newly added claims are also patentable over the cited documents at least for the reasons noted above. Support for this claim can be found within the specification and drawings as originally filed, thus no new matter was added.

Applicant has made a diligent and sincere effort to place this application in condition for immediate allowance and notice to this effect is earnestly solicited. To expedite prosecution of this application to allowance, the Examiner is invited to call the undersigned attorney to discuss any issues relating to this application.

Early and favorable action is earnestly solicited.

Respectfully submitted,
Arun K. Gupta

Date: October 31, 2002

Michael K. Kinney, Reg. No. 42,740
Attorney for Applicant

Direct: (203) 498-4411
mkinney@wiggin.com

WIGGIN & DANA LLP
One Century Tower
P.O. Box 1832
New Haven, Connecticut 06508-1832

Telephone: (203)498-4400

Facsimile: (203) 782-2889

\12976\14\369840.1

## MARKED-UP VERSION OF CLAIMS

1. (Amended) A method for extracting desired data [from a digital database] for metric analysis, the method comprising the steps of:

a. [Specifying] specifying desired data elements to be extracted from an operational database;

b. generating executable code from said [identified] specified data elements for extracting said [identified] specified data elements from said operational database; and

c. executing said executable code thereby extracting said [identified] specified data elements from said operational database and storing said extracted data in a staging database.

2. (Amended) The method of claim 1 wherein said [executable code stores the extracted data elements in a staging database] step of specifying includes identifying attributes and processes related to said desired data elements.

3. (Amended) The method of claim [2] 1 wherein said operational database is a relational database.

4. (Amended) The method of claim [3] 1 wherein said staging database is a relational database.

5. (Amended) The method of claim [4] 2 wherein identifying the data elements, attributes and processes [to be extracted comprises the steps of selecting data elements stored in a repository through] is performed utilizing a graphical user interface.

Please add the following new claims.

-- 6. The method of claim 5 wherein the step of identifying includes deriving rules for extracting said desired data elements from graphical representations and manually defined values of said attributes and processes related to said desired data elements.

7. The method of claim 1 wherein the step of executing includes invoking a background process for extracting and storing said specified data elements.

8. A method for developing a metric analysis profile and retrieving data elements of interest in accordance with the metric analysis profile, comprising:

modeling business activities as interactions and relationships between a plurality of business classes, wherein a business class includes attributes, processes and sub-processes corresponding to data elements of the business activities;

storing the plurality of business classes and the data elements in an operational database;

selectively specifying attributes, processes, sub-processes and data elements for inclusion within the metric analysis profile;

automatically generating executable code from the specified attributes, processes, sub-processes and data elements for retrieving the specified data elements from the operational database and for storing the specified data elements in a staging database.

9. The method of claim 8, upon receipt of a request to delete one of the plurality of business classes, the method further comprising:

automatically generating executable code for deleting the specified attributes, processes and sub-processes corresponding to the requested business class; and

when at least one of the plurality of business classes depends from the requested business class, propagating the generated code to delete the at least one dependent business class.

10. The method of claim 9 wherein the code for deleting includes code for de-allocating memory storage space in the operational database associated to the deleted business class.

11. The method of claim 8 wherein the request for deletion is automatically generated in response to a change in the metric analysis profile.

12. The method of claim 8 wherein the step of storing in the operational database includes storing the plurality of business classes and data elements as a repository based aggregration in accordance with the modeled interactions and relationships.

13. A system for developing a metric analysis profile and retrieving data elements of interest in accordance with said metric analysis profile, comprising:

a graphical user interface for modeling business activities as interactions and relationships between a plurality of business classes, wherein a business class includes attributes, processes and sub-processes corresponding to data elements of said business activities, said graphical user interface for selectively specifying attributes, processes, sub-processes and data elements for inclusion within said metric analysis profile;

an operational database for storing said plurality of business classes and said data elements;

a staging database; and

executable code automatically generated from said specified attributes, processes, sub-processes and data elements, said code for retrieving said specified data elements from said operational database and for storing said data elements in said staging database for analysis thereon.

14. The system of claim 13, further comprising executable code automatically generated upon receipt of a request to delete one of said plurality of business classes, said executable code for deleting said specified attributes, processes and sub-processes corresponding to said requested business class.

15. The system of claim 14 wherein when at least one of said plurality of business classes depends from said requested business class, said executable code includes logic for propagating to said at least one depend business class and deleting said at least one dependent business class.

16. The system of claim 14 wherein said request for deletion is automatically generated in response to a change in said metric analysis profile. --